Panel: Trust Management: a Simple, Scaleable Approach to Internet Client Security - or Is It?!

Barbara Fox, Chair bfox@microsoft.com
Security Architect
Microsoft Corporation One Microsoft Way Redmond, WA 98052

Security and privacy in this new generation of the Internet has to be rethought. Traditional network security focuses on protecting valuable server resources from hostile clients. But now the problem also includes protecting valuable *client* resources from hostile servers. Information is flowing in both directions between publishers and consumers of Internet content, and it's going to take collaboration between them to make any real difference in Internet security. Inevitably publishers and consumers must choose at some point to *trust* each other.

Trust in this context is a product of intuition, experience, and information – with an Internet twist. Ultimately, the idea is not to avoid all potentially dangerous operations; burying our heads in the sand only reduces the usefulness of the network. Instead, we need to explicitly acknowledge that an action *is* potentially dangerous and provide the user with tools to make an informed decision about whether to permit that action to occur. . This is *trust management*.

In its simplest form, trust in mobile code can be associated with the digital signature of its publisher. In practice, however, this "shrink-wrap equivalent" scheme quickly fell victim to the success of the web. Anybody can sign native code, ignore CA policy, and wreak havoc.

The Java "sandbox" model, in which classes loaded from the network were granted extremely limited capabilities while classes loaded from the local disk were given free reign to do virtually anything, also missed the mark. Without any means of "opting-out" of any of the sandbox restrictions individual servers cannot deploy an entire class of applications their clients desire. Currently, Netscape, Sun and Microsoft have all added intermediate levels of trust to their security models to provide "opt-out" mechanisms. Enhanced administrative options for the virtual machine include fine-grained control over particular capabilities granted to Java code (e.g. access to scratch space, local files, and network connections). This model allows an application to be given *some* additional capabilities, but under tightly controlled conditions.

Unfortunately, this increase in functionality has brought along with it an explosion of complexity. What was previously a simple idea - "Don't let bad things happen on or to my machine." - has become a complex web of interrelated choices for the user. And users are no longer "just surfing" passively. Corporations have just recently realized that the combination of a Web Browser and HTML can produce powerful internal applications. For the user though, it's not reasonable to allow all content the same access to his system. So, the user browses to Site A. The browser then dutifully asks if Site A can: *Write to the Disk, Perform Network Operations, Run ActiveX Controls, Script ActiveX controls, Use more than 1MB of memory,* and *Perform Cross Frame Operations.*

Huh? Are these questions that most users can answer? Not likely – at least with any measure of confidence. The most obvious way to reduce this inherently dangerous user interface clutter is to let administrators pre-configure (and, of course, dynamically update) desktops to discriminate based on source URL, digital signature and, for Java, specific privileges requested. For home users, a simple set of defaults can be "baked into the application."

But does this mean we're done? Or, since Microsoft, Netscape, and Sun have all taken slightly different approaches to how trust gets managed, have we just created more confusion? It certainly appears that developers have to make some choices on how they package their components for Internet delivery --- but have we all gone too far on the complexity curve for our customers?

This panel will focus specifically on the ActiveX and Java security models: what they have in common, where they differ, and what underlying security principles they're built on. Further, they will be asked to predict whether we will eventually find ourselves evolving or junking this whole approach to Internet client security.